

C

博客

登录 | 注册

Q

≡

个人资料



v\_JULY\_v

+ 加关注

✉ 发私信



☰

恒

MVP

★

访问： 11261467次

积分： 46776

等级： 

BLOG > 8

排名： 第47名

-----

原创： 151篇 转载： 0篇

译文： 5篇 评论： 13276条

博主简介

July，于2010年10月11日开始在CSDN上写博（搜索：“结构之法”，进入本博客），博客专注面试、算法、机器学习。2015年正式创业，七月在线科技创始人兼CEO，公司官网：七月在线 (<https://www.julyedu.com/>)，微博@研究者July。新书《编程之法》15年10月14日起正式上市。JulyEdu c/c++/算法Q群：123531805。July，2016/5月。

我的微博

💡 [【公告】博客系统优化升级](#) [【收藏】Html5 精品资源汇集](#) [我们为什么选择Java](#)

📄 原

程序员编程艺术第二十五章：Jon Bentley：90%无法正确实现二分查找

标签： [编程](#) [测试](#) [chm](#) [算法](#) [blog](#) [文档](#)

2011-12-21 19:35 🔍 64124人阅读 💬 评论(122) [收藏](#) [举报](#)

☰ 分类：

[19.TAOPP algorithms \(7\)](#) ▼ [11.TAOPP（编程艺术） \(33\)](#) ▼ [13.TAOPP array \(13\)](#) ▼

目录(?)	[+]
目录(?)	[+]

第二十五章：二分查找实现（Jon Bentley：90%程序员无法正确实现）

作者：July

出处：结构之法[算法](#)之道

引言

Jon Bentley：90%以上的程序员无法正确无误的写出二分查找代码。也许很多人都早已听说过这句话，但我还是想引用《编程珠玑》上的如下几段文字：

“二分查找可以解决（[预排序数组的查找](#)）问题：只要数组中包含T（即要查找的值），那么通过不断缩小包含T的范围，最终就可以找到它。一开始，范围覆盖整个数组。将数组的中间项与T进行比较，可以排除一半元素，范围缩小一半。就这样反复比较，反复缩小范围，最终就会在数组中找到T，或者确定原以为T所在的范围实际为空。对于包含N个元素的表，整个查找过程大约要经过log(2)N次比较。

多数程序员都觉得只要理解了上面的描述，写出代码就不难了；但事实并非如此。如果你不认同这一点，最好的办法就是放下书本，自己动手写一写。试试吧。

我在贝尔实验室和IBM的时候都出过这道考题。那些专业的程序员有几个小时的时间，可以用他们选择的语言把上面的描述写出来；写出高级伪代码也可以。考试结束后，差不多所有程序员都认为自己写出了正确的程序。于是，我们花了半个钟头来看他们编写的代码经过测试用例验证的结果。几次课，一百多人的结果相差无几：90%的程序员写的程序中有bug（我并不认为没有bug的代码就正确）。

我很惊讶：在足够的时间内，只有大约10%的专业程序员可以把这个小程序写对。但写不对这个小程序的还不止这些人：高德纳在《计算机程序设计的艺术 第3卷 排序和查找》第6.2.1节的“历史与参考文献”部分指出，虽然早在1946年就有人将二分查找的方法公诸于世，但直到1962年才有人写出没有bug的二分查找程序。”——乔恩·本特利，《编程珠玑（第1版）》第35-36页。

你能正确无误的写出二分查找代码么？不妨一试。

二分查找代码

二分查找的原理想必不用多解释了，不过有一点必须提醒读者的是，二分查找是针对的排好序的数组。OK，纸上读来终觉浅，觉知此事要躬行。我先来写一份，下面是我写的一份二分查找的实现（之前去某一家公司面试也曾被叫当场实现二分查找，不过结果可能跟你一样，当时就未能完整无误写出），有任何问题或错误，恳请不吝指正：

July新书《编程之法》上市



京东 当当 天猫 Amazon  
异步社区 互动出版网

文章分类

- 03.Algorithms（实现）（9）
- 01.Algorithms（研究）（27）
- 02.Algorithms（后续）（22）
- 04.Algorithms（讨论）（1）
- 05.MS 100' original（7）
- 06.MS 100' answers（13）
- 07.MS 100' classify（4）
- 08.MS 100' one Keys（6）
- 09.MS 100' follow-up（4）
- 10.MS 100' comments（4）
- 11.TAOPP（编程艺术）（34）
- 12.TAOPP string（8）
- 13.TAOPP array（14）
- 14.TAOPP list（2）
- 15.stack/heap/queue（0）
- 16.TAOPP tree（2）
- 17.TAOPP c/c++（2）
- 18.TAOPP function（2）
- 19.TAOPP algorithms（8）
- 20.number operations（1）
- 21.Essays（7）
- 22.Big Data Processing（5）
- 23.Redis/MongoDB（0）
- 24.data structures（12）
- 25.Red-black tree（7）
- 26.Image Processing（3）
- 27.Architecture design（4）
- 28.Source analysis（3）
- 29.Recommend&Search（4）
- 30.Machine L & Deep Learning（12）

```
//二分查找V0.1实现版
//copyright@2011 July
//随时欢迎读者找bug, email: zhoulei0907@yahoo.cn。

//首先要把握下面几个要点:
//right=n-1 => while(left <= right) => right=middle-1;
//right=n  => while(left < right) => right=middle;
//middle的计算不能写在while循环外, 否则无法得到更新。

int binary_search(int array[],int n,int value)
{
    int left=0;
    int right=n-1;
    //如果这里是int right = n 的话, 那么下面有两处地方需要修改, 以保证一一对应:
    //1、下面循环的条件则是while(left < right)
    //2、循环内当array[middle]>value 的时候, right = mid

    while (left<=right)           //循环条件, 适时而变
    {
        int middle=left + ((right-left)>>1); //防止溢出, 移位也更高效。同时, 每次循环都需要更新。

        if (array[middle]>value)
        {
            right =middle-1;    //right赋值, 适时而变
        }
        else if(array[middle]<value)
        {
            left=middle+1;
        }
        else
            return middle;
        //可能会有读者认为刚开始时就要判断相等, 但毕竟数组中不相等的情况更多
        //如果每次循环都判断一下是否相等, 将耗费时间
    }
    return -1;
}
```



收藏到代码笔记



收藏到代码笔记

[cpp] view plain copy print ?

```
//二分查找V0.1实现版
//copyright@2011 July
//随时欢迎读者找bug, email: zhoulei0907@yahoo.cn。

//首先要把握下面几个要点:
//right=n-1 => while(left <= right) => right=middle-1;
//right=n  => while(left < right) => right=middle;
//middle的计算不能写在while循环外, 否则无法得到更新。

int binary_search(int array[],int n,int value)
{
    int left=0;
    int right=n-1;
    //如果这里是int right = n 的话, 那么下面有两处地方需要修改, 以保证一一对应:
    //1、下面循环的条件则是while(left < right)
    //2、循环内当array[middle]>value 的时候, right = mid

    while (left<=right)           //循环条件, 适时而变
    {
        int middle=left + ((right-left)>>1); //防止溢出, 移位也更高效。同时, 每次循环都需要更新。

        if (array[middle]>value)
        {
            right =middle-1;    //right赋值, 适时而变
        }
        else if(array[middle]<value)
        {
            left=middle+1;
        }
        else
            return middle;
        //可能会有读者认为刚开始时就要判断相等, 但毕竟数组中不相等的情况更多
        //如果每次循环都判断一下是否相等, 将耗费时间
    }
    return -1;
}
```

收藏到代码笔记

简单测试下, 运行结果如下所示（当然, 一次测试正确不代表程序便0 bug了, 且测试深度远远不够）：



博客专栏



数据挖掘十大算法系列

文章：12篇

阅读：1415314



微软面试100题系列

文章：18篇

阅读：2875834



程序员编程艺术

文章：32篇

阅读：2207236



经典算法研究

文章：32篇

阅读：2837224

文章搜索



阅读排行

支持向量机通俗导论（理	(491245)
程序员面试、算法研究、	(481215)
教你如何迅速秒杀掉：99	(434334)
从B树、B+树、B*树谈到	(350464)
九月十月百度人搜，阿里	(268826)
十道海量数据处理面试题	(267380)
十一、从头到尾解析Hash	(226403)
从头到尾彻底理解KMP	(218750)
横空出世，席卷互联网--i	(211980)
教你初步了解红黑树	(190798)

评论排行

程序员面试、算法研究、	(495)
支持向量机通俗导论（理	(490)
九月十月百度人搜，阿里	(390)
从头到尾彻底理解KMP	(357)
九月腾讯，创新工场，淘	(351)
从B树、B+树、B*树谈到	(348)
当今世界最为经典的十大	(327)
横空出世，席卷互联网--i	(292)
教你如何迅速秒杀掉：99	(281)
程序员编程艺术：第一章	(256)

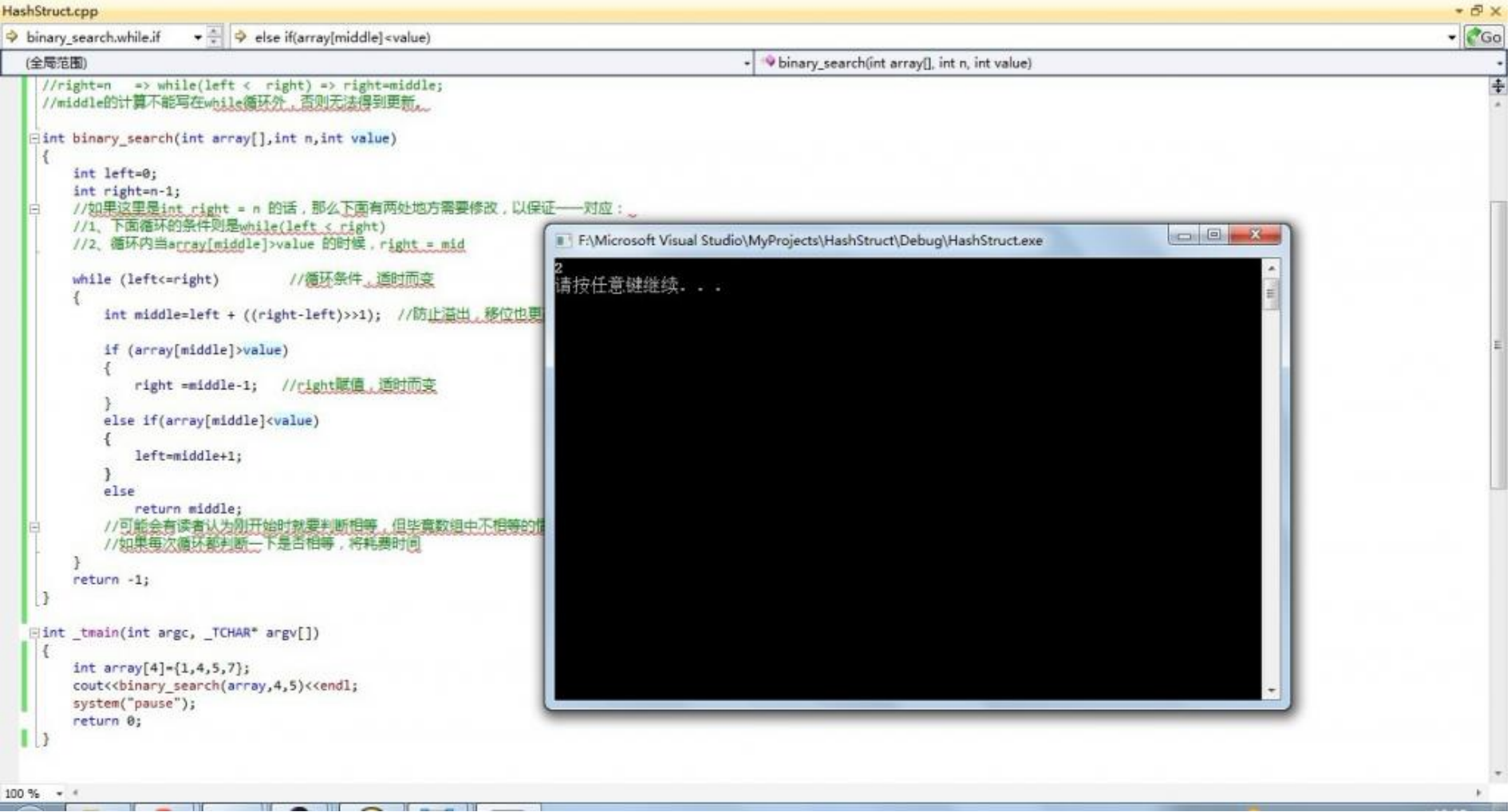
最新评论

支持向量机通俗导论（理解SVM

sunseazhu: @supercopyman:很赞同你的理解，自己也在这边倒腾了很久。简单理解：因为超平面只与法向量和...

CNN笔记：通俗理解卷积神经网络

kenny\_star: 对于特征图28\*28大小，用5\*5的滤波器去卷积。从C1层到S2层，特征图大小



## 测试

也许你之前已经把二分查找实现过很多次了，但现在不妨再次测试一下。关闭所有网页，窗口，打开记事本，或者编辑器，或者直接在本文评论下，不参考上面我写的或其他任何人的程序，给自己十分钟到N个小时不等的时间，立即编写一个二分查找程序。独立一次性正确写出来后，可以留下代码和邮箱地址，我给你传一份本blog的博文集锦CHM文件 && 十三个经典**算法**研究带标签+目录的PDF文档（你也可以去我的资源下载处下载：[http://download.csdn.net/user/v\\_july\\_v](http://download.csdn.net/user/v_july_v)）。


当然，能正确写出来不代表任何什么，不能正确写出来亦不代表什么，仅仅针对Jon Bentley的言论做一个简单的测试而已。下一章，请见第二十六章：基于给定的文档生成倒排索引的编码与实践。谢谢。

## 总结


本文发表后，马上就有很多朋友自己尝试了。根据从朋友们在本文评论下留下的代码，发现**出错率最高的在以下这么几个地方**：

1. 注释里已经说得很明白了，可还是会有不少朋友犯此类的错误：

```
//首先要把握下面几个要点：
//right=n-1 => while(left <= right) => right=middle-1;
//right=n   => while(left <  right) => right=middle;
//middle的计算不能写在while循环外，否则无法得到更新。
```



收藏到代码笔记



收藏到代码笔记

2. 还有一个最最常犯的错误是@土豆：  
middle= (left+right)>>1; 这样的话left与right的值比较大的时候，其和可能溢出。

各位继续努力。  
updated：各位，可以到此处0积分下载本blog最新博文集锦第6期CHM文件：[http://download.csdn.net/detail/v\\_july\\_v/4020172](http://download.csdn.net/detail/v_july_v/4020172)。



顶

55

踩

12

CNN笔记：通俗理解卷积神经网络：HLhacker1: 学到了

支持向量机通俗导论（理解SVM）躺在草原看夕阳: @v\_JULY\_v:@victor0535:我的理解：1. 目标是  $\max(r^*/||w||)$ ；...

支持向量机通俗导论（理解SVM）躺在草原看夕阳: @victor0535:我的理解：1. 目标是  $\max(r^*/||w||)$ ；需要要求的参数为w和b...

CNN笔记：通俗理解卷积神经网络：Eyecools: 好文章，膜拜一下

CNN笔记：通俗理解卷积神经网络：Eyecools: 好文章，膜拜一下

红黑树从头至尾插入和删除结点f\_\_zourun\_\_: 请问删到全剩黑色节点的时候，再去删除13这个节点的时候怎么删除的，看你代码好像和这个过程不一样啊

B树的C实现  
阳光梦: 您好，测试代码开源下吧。

CNN笔记：通俗理解卷积神经网络：v\_JULY\_v: @Mark\_LQ:你好，非常感谢你的提醒，之前的图确实有问题，所以刚刚根据cs231n的卷积动图依次...

Google or baidu?

Google搜--"结构之法"（My BLOG）

baidu 搜--"结构之法"（My BLOG）

我的驻点

- 00、我的新浪微博
- 01、我的Github主页
- 02、七月在线
- 03、寒
- 04、joycewyj的机器学习笔记
- 05、Harry
- 06、NoSQLFan
- 07、酷勤网
- 08、52nlp
- 09、IT面试论坛
- 10、北大朋友的挖掘乐园
- 11、跟Sophia\_qing一起读硕士
- 12、caopengcs
- 13、51nod
- 14、韩寒
- 15、曾经的叛逆与年少
- 16、code4app:iOS代码示例
- 17、斯坦福机器学习公开课
- 18、Memory Model与并发编程
- 19、淘宝搜索技术博客
- 20、interviewstreet
- 21、LeetCode
- 22、Team\_Algorithms

文章存档

- 2016年07月（1）
- 2015年10月（1）
- 2015年08月（1）
- 2014年11月（4）
- 2014年10月（1）

我的同类文章

19.TAOPP algorithms（7）	11.TAOPP（编程艺术）（33）	13.TAOPP array（13）
<ul style="list-style-type: none"><li>程序员编程艺术第三十六~...2013-09-07 阅读 38546</li><li>编程艺术第十六~第二十章...2011-10-17 阅读 52499</li><li>程序员编程艺术：三之三续...2011-05-28 阅读 32541</li><li>程序员编程艺术：第三章续...2011-05-08 阅读 62428</li></ul>		<ul style="list-style-type: none"><li>程序员编程艺术第一~二十...2011-11-23 阅读 53874</li><li>程序员编程艺术第十一章：...2011-08-17 阅读 46115</li><li>十四、第三章再续：快速选...2011-05-19 阅读 40547</li></ul>

参考知识库



算法与数据结构知识库

2235 关注 | 3973 收录

猜你在找

- PDF神器-Adobe Acrobat Pr
- 软件测试工程师面试前突击——100道试题精讲视频（面
- 软件测试基础
- 数据结构和算法
- HTML 5移动开发从入门到精通
- 软件测试工程师面试前突击——100道试题精讲视频（面
- PDF神器-Adobe Acrobat Pr
- 软件测试基础
- 数据结构和算法
- 韦东山嵌入式Linux第一期视频
- 90无法正确实现二分查找
- 二分查找的三个版本据说90的计算机专家不能在2小时内
- 技术文档《编程珠玑》Jon Bentley • 第9章 代码调优
- 技术文档《编程珠玑》Jon Bentley • 第2章 啊哈算法

查看评论

84楼 qq906237201 2016-04-25 17:10发表



博主看看我的

```
unsigned int fun(const ARRY &data, const unsigned int low, const unsigned int high, const int target)
{
    //ARRAY 是 int [1000]类型，用typedef声明的
    //low和high分别是数组的下界和上界
    //CEIL和FLOOR相当于你的right和left
    unsigned int ceil{ high }, floor{low};
    unsigned int index{ floor + ((ceil - floor)>>1) };
    while (floor <= ceil)
    {
        if (target == data[index])return index;
        if (target == data[floor])return floor;
        if (target == data[ceil])return ceil;
        else
        {
            if (target > data[index])
            {
                floor = index + 1;
                ceil = ceil - 1;
            }
            else
            {
                ceil = index - 1;
                floor = floor + 1;
            }
        }
    }
}
```

展开

```
index = floor + ((ceil - floor) >> 1);
}
return 0;

}
```

83楼 [liango](#) 2016-01-13 18:34发表



再来一个

```
```scala
def binarySearch(arr: Array[Double], target: Double): Int = {
def go(left: Int, right: Int): Int = {
val mid = left + ((right - left) >> 1)
println(s"arr($mid) = $arr(mid), target = $target")
arr(mid) match {
case v if v == target => mid
case v if v > target => go(left, mid - 1)
case v if v < target => go(mid + 1, right)
}
}
go(0, arr.length - 1)
}
```
```

82楼 [liango](#) 2016-01-13 18:12发表



```
```scala
object Main extends App {

def binary_search(arr: Array[Double], target: Double): Int = {
def go(arr: Array[Double], left: Int, right: Int): Int = {
val mid = left + ((right - left) >> 1)
if (arr(mid) == target) mid
else if (arr(mid) > target) go(arr, left, mid - 1)
else go(arr, mid + 1, right)
}
go(arr, 0, arr.length - 1)
}

// test
val arr = Array[Double](0.0, 1.0, 2.0, 3.0, 4.0, 5.0)
val pos = binary_search(arr, 3.0)
println("pos = " + pos)
}
```
```

81楼 [刘蚊子](#) 2015-11-09 16:36发表



```
function binary_search(&$sortedList, $targetValue, $targetPos=null, $unlookedMin=null, $unlookedMax=null)
{
    $unlookedMin = $unlookedMin==null ? 0 : $unlookedMin;
    $unlookedMax = $unlookedMax==null ? sizeof($sortedList) : $unlookedMax;
    $targetPos = $targetPos==null ? intval($unlookedMax/2) : $targetPos;
    if($unlookedMax-$unlookedMin<2){
        if($sortedList[$unlookedMin] == $targetValue){return $unlookedMin;}
        elseif($sortedList[$unlookedMax] == $targetValue){return $unlookedMax;}
        else{return -1;}
    }
    if($sortedList[$targetPos] == $targetValue){return $targetPos;}
    elseif($sortedList[$targetPos] < $targetValue){
        $nextTargetPos = $targetPos + intval(($unlookedMax-$targetPos)/2);
        return binary_search($sortedList, $targetValue, $nextTargetPos, $targetPos, $unlookedMax);
    }
    elseif($sortedList[$targetPos] > $targetValue){
        $nextTargetPos = $unlookedMin + intval(($targetPos - $unlookedMin)/2);
        return binary_search($sortedList, $targetValue, $nextTargetPos, $unlookedMin, $targetPos);
    }
}
```

80楼 [逆风微积分](#) 2015-05-11 16:53发表



4月份面试时写的二分查找算法，博主上面列出来的容易出错的地方都在我身上发生了

79楼 [逆风微积分](#) 2015-05-11 16:52发表





4月份面试时写的二分查找算法， 博主上面列出来的容易出错的地方都在我身上发生了

78楼 [小木偶的怪脾气](#) 2015-04-17 14:25发表



```
望楼主帮我审批一下：（万分感谢）

#include <iostream>
#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <set>
#include <algorithm>

#define MAXN 1010
#define RST(N)memset(N, 0, sizeof(N))
using namespace std;

int cmp(const void *a, const void *b)
{
    return *(int *)a - *(int *)b;
}

bool Bin_Search(const int *pre, int low, int high, const int key)
{
    high--;
    while(low <= high) {
        int mid = (low + high) >> 1;
        if(pre[mid] == key) return true;
        else if(pre[mid] < key) low = mid + 1;
        else high = mid - 1;
    }
    return false;
}

int main()
{
    int n, m, a[MAXN], Mc;
    while(cin >> n >> m) {
        for(int i=0; i<n; i++) cin>>a[i];
        qsort(a, n, sizeof(int), cmp);
        while(m--) {
            cin >> Mc;
            bool flag = Bin_Search(a, 0, n, Mc);
            flag ? cout << "YES\n" : cout << "NO\n";
        }
        return 0;
    }
```

77楼 [轩辕夏禹](#) 2014-09-11 18:37发表



“int middle=left + ((right-left)>>1);”  
当(right-left)最高位为1时，是不是会发生符号扩展？

Re: [czzshr](#) 2014-09-12 20:03发表



回复nkguohao：int 是有符号整数， 所以右移是逻辑右移， 不会出问题的。  
我估计楼主之所以那么写而不是像我那样写， 主要是为了提高计算范围：有些情况按我的写法会溢出， 按楼主的写法则不会。

76楼 [czzshr](#) 2014-08-29 19:33发表



int mid = left + ((right - left) >> 1);  
明明可以写成  
int mid = (left + right) >> 1;

Re: [蝶变\\_流年](#) 2014-09-10 14:42发表



回复mxdllove00：不可以的，比如说：把int看作char（有符号），low和high分别是1和7，那么1+7得-8，此时进行右移需注意：C语言中执行算术右移还是逻辑右移不同的机器执行的操作不一样，如果是逻辑右移那么程序正确，而执行算术右移（移出空位补符号位）最终得结果为-4，显然不行。而low+（high-low）>> 1 不会出现这种情况。。。其实关于这种进制的问题最好看看CSAPP第2章，绝对涨姿势。。。。

Re: [czzshr](#) 2014-09-10 19:33发表



回复u010518493: "1+7得-8" 什么意思?

Re: 蝶变\_流年 2014-09-10 19:46发表



回复mxdllove00: 呃, 写错了, 本来想写byte类型来, 细一想byte也不对: ), 我这里假设的是四个位的数据类型, 就是一个16进制位, 假设是有符号的。所以0111+0001=1001, 即-8。

75楼 zizi7 2014-08-29 16:29发表



貌似楼主没有考虑被查找数有重复的情况, 贴个我的代码, 还请指正!

```
void halffind(int* data,int value,int start,int end,int *find)
{
    if(end-start<=0)
        return;

    int half=(end-start)/2+start;
    if(data[half]==value)
    {
        find[++find[0]]=half;
        //检查是否有重复
        int flag=half;
        while(data[++flag]==value)
        {
            find[++find[0]]=flag;
            if(flag==end-1)
                break;
        }
        flag=half;
        while(data[--flag]==value)
        {
            find[++find[0]]=flag;
            if(flag==start)
                break;
        }
        return;
    }
    else if(data[half]>value)
        halffind(data,value,start,half,find);
    else
        halffind(data,value,half+1,end,find);
}
```

74楼 李嘉图mTONG 2014-08-08 17:33发表



还有一点是不是该注意一下 while循环内 如果left = mid的话会造成死循环 {1 2 3 4 5} left 指向4 right 指向5 查找5的时候的时候会造成死循环 left可能永远等于mid 但是 right可以等于mid 不一定是mid+1 并不会造成死循环 - 我觉得这也是该注意的地方

Re: 李嘉图mTONG 2014-08-08 17:34发表



回复u013479704: right 不一定是 mid -1 上面写错了

73楼 hyhai7 2014-07-15 16:35发表



我最欣赏的一行代码,很明显解决了数值溢出问题

```
int middle=left + ((right-left)>>1);
```

但是也有不足之处, int middle 应该声明在循环体外面,

挑错很容易,写很难,多谢了,博主

72楼 RowandJJ 2014-07-11 10:48发表



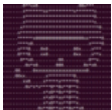
大神, 我觉着是不是可以把Middle的声明放在while外面, 因为你每次while循环都重新声明定义了middle.....

Re: vistastar2 2014-11-05 20:40发表



回复RowandJJ: 不会重复定义, 它只是同一个变量。

71楼 禾斗人 2014-04-15 22:34发表



看这边界条件, 总纳闷为什么是这样的,

```
//right=n-1 => while(left <= right) => right=middle-1;
```

```
//right=n => while(left < right) => right=middle;
```

除了直观的理解, 感觉可以这样分析:

因为要跳出while循环, 避免死循环。

因此, 当left <= right时, right = middle - 1 = (left + right)/2 - 1 => left = right + 2 ,从而可以跳出循环。

当left < right时, right = middle = (left + right) / 2 => left = right, 同样可以跳出循环而不会出现死循环。

请lz鉴别是否有理~~

70楼 [miraclecoder](#) 2014-04-10 16:12发表 



不太清楚：


//首先要把握下面几个要点：

//right=n-1 => while(left <= right) => right=middle-1;

//right=n => while(left < right) => right=middle;

//middle的计算不能写在while循环外，否则无法得到更新。

但是如果right赋值为n, 则很有可能最后两个数，其中right为n,应该比较特殊。但还是未懂原因，还请不吝赐教

69楼 [beef9999](#) 2013-11-02 22:42发表 



```
include <iostream>
using namespace std;
const int LEN = 100;

int *bsearchWrapper(int *p, int size, int obj);


int main()
{
    int a[LEN];
    for(int i = 0; i < LEN; ++i)
        a[i] = 3 * i;

    int *dst = bsearchWrapper(a, LEN , 123);
    cout<<dst<<endl;
    if(dst)
        cout<<*dst<<endl;
}

void bsearch(int *p, int begin, int end, int obj, int **ppRet)
{
    if(p == NULL || begin < 0 || end >= LEN || begin > end)
        return;
    int mid = begin + (end - begin) / 2;
    int mid_obj = p[mid];
    if(mid_obj < obj)
        bsearch(p, mid + 1, end, obj, ppRet);
    else if(mid_obj > obj)
        bsearch(p, begin, mid - 1, obj, ppRet);
    else if(mid_obj == obj)
        *ppRet = &p[mid];
    else return;
}
```

楼上的都不考虑查找不到的情况吗？你们提供的这个函数别人根本没法用

还有返回-1的，如果别人数字里就有-1怎么办

Re: [panpan\\_117](#) 2013-11-17 20:37发表 



回复beef9999：返回-1有什么关系。程序中返回的是值在数组中位置。位置怎么可能是-1.

68楼 [YanqingYoung](#) 2013-09-08 16:59发表 



```
//考虑有相同值的非递减二分查找
template <class T> //T 需要重载 == <=
int binarySearch(const T a[],int n,const T& t)
{
    if (a == NULL || n<1)
    {
        return -1;
    }

    int nLeft = 0;
    int nRight = n - 1;
    int nMiddle;

    while (nLeft<nRight-1)//nLeft 等于 nRight-1 or nRight
    {
        nMiddle = nLeft + ((nRight-nLeft)>>1);

        //if (a[nMiddle]<=t)//这里取相同值的最大下标，取最小时去掉等于号
        if (a[nMiddle]<t)//这里取相同值的最小下标，取最大时加上等于号
        {
            nLeft = nMiddle;
        }
        else
        {
            nRight = nMiddle;
        }
    }

    // if (a[nRight]==t) //这里取最大下标 取最小时先取nLeft
    // {
    //     return nRight;
    // }
    // else if (a[nLeft] == t)
    // {
    //     return nLeft;
    // }
```



```
if (a[nLeft]==t) //这里取最大下标 取最大时先取nRight
{
    return nLeft;
}
else if (a[nRight] == t)
{
    return nRight;
}
else
{
    return -1;
}
}

int main()
{
    int a[]={4,4,5,6,7,7,9,42};

    cout<<binarySearch(a,8,0)<<endl;

    return 0;
}
```

Re: [kxcfzyk](#) 2014-12-27 13:36发表



回复YanqingYoung: STL里面的lower\_bound/upper\_bound写法更加简洁一些:

```
//返回第一个大于等于t的位置, 可能为n, -1表示错误
template <class T> //T需要重载<或<=
int binarySearch(const T a[], int n, const T& t) {
    if (a == NULL || n < 1) {
        return -1;
    }
    int nLeft = 0, nRight = n;
    while (nLeft < nRight) {
        int nMiddle = nLeft + ((nRight - nLeft) >> 1);

        //此处 <= 最终得到第一个大于t的位置
        if (a[nMiddle] < t) {
            nLeft = nMiddle + 1;
        } else {
            nRight = nMiddle;
        }
    }
    return nLeft;
}
```

Re: [donshing](#) 2013-10-14 15:25发表



回复YanqingYoung: 这位兄台的编程思想和我一摸一样

67楼 [bigwangdi](#) 2013-09-06 14:32发表



不用对输入做检查的吗?  
输入空指针呢?

66楼 [iegyiy](#) 2013-07-31 17:30发表



```
public static int dichotomy(int[] a, int start, int end, int m) {
    int mid = (end - start) >>> 1;
    int v = a[mid];
    while (v != m && (mid > start && mid < end)) {
        if (v > m) {
            end = mid;
            mid = ((end - start) >>> 1) + start;
            v = a[mid];
        }
        if (v < m) {
            start = mid;
            mid += ((end - start) >>> 1);
            v = a[mid];
        }
    }
    if (v != m) {
        return -1;
    }
    return mid;
}
```

65楼 [paopaocaoren](#) 2013-06-29 20:12发表



```
#include <iostream>
#include <stdlib.h>
#include <ctime>
```

```

#include <algorithm>
#define SIZE 100
using namespace std;
int binary_search(int A[],int p,int q,int key)
{
    int mid;
    if(p<=q)
    {
        mid=(p+q)/2;
        if(A[mid]==key)
            return mid;
        else if(key<A[mid])
        {
            return binary_search(A,p,mid-1,key);
        }
        else
        {
            return binary_search(A,mid+1,q,key);
        }
    }
    return -1;
}
int main()
{
    int key,i,t;
    int num[SIZE];
    srand((unsigned)time(NULL));
    for(i=0;i<SIZE;i++)
    {
        num[i]=rand()%10000;
    }
    sort(num,num+SIZE);
    cout<<"一共有"<<SIZE<<"个数据"<<endl;
    for(i=0;i<SIZE;i++)
    {
        if((i+1)%15==0)
        {
            cout<<num[i]<<" "<<endl;
        }
        else
            cout<<num[i]<<" ";
    }
    cout<<endl;
    cout<<"请输入要搜索的关键值,退出请输入-999: "<<endl;
    while(cin>>key && key!=-999)
    {
        t=binary_search(num,0,SIZE-1,key);
        if(t!=-1)
            cout<<"关键值不再这堆数据中"<<endl;
        else
        {
            cout<<"关键值的位置是"<<t+1<<endl;
        }
        cout<<"-----"<<endl;
        cout<<"请输入要搜索的关键值,退出请输入-999: "<<endl;
    }
    return 0;
}

```

64楼 [coderchenjingui](#) 2013-06-03 10:32发表



```

int binary_search(int a[], int num, int find)
{
    int left = 0;
    int right = num - 1;
    int mid;
    while(right >= left){
        mid = (left + right) / 2;
        if(a[mid] > find)
            right = mid - 1;
        else if(a[mid] < find)
            left = mid + 1;
        else
            return mid;
    }
    return -1; //not find
}

```

唉，我也是bug制造者。

63楼 [natulis](#) 2013-04-14 17:14发表



发现个小细节问题，left + ((right-left)>>1)  
位移的优先级比四则运算低，所以不用写这么多括号，我觉得写成left + (right - left >> 1)就行了，博主你看看行不

62楼 [zhaojia1989](#) 2013-03-26 16:00发表



```

#include <iostream>
using namespace std;

```

```
int binary_search(int a[], int length, int dst)
{
    int *p = a;
    while (length != 0)
    {
        if (p[length/2] == dst)
            return length/2;
        if (p[length/2] > dst)
            length /= 2;
        else
        {
            length /= length;
            p += length;
        }
    }
    return -1;
}

int main()
{
    int a[] = {1,2,3,4};
    int rst = binary_search(a, 4, 3);
    cout << rst << endl;
    return 0;
}

为什么不考虑操作指针？
```

61楼 [onehao](#) 2013-02-20 22:12发表



今天来跟着楼主一起学习，谢谢！

60楼 [syzcch](#) 2012-10-16 13:15发表



确实 写出没bug的程序不容易

59楼 [slayerxj](#) 2012-10-12 14:16发表



```
int binary_search(int array[], int n, int value)
{
    if ((array == NULL) || (n == 0)) {
        return -1;
    }
    if (array[0] == value) {
        return 0;
    }
    if (array[n] == value) {
        return n;
    }

    int left = 0;
    int right = n;
    int middle = n >> 1;
    while (right - left != 1) {
        if (value < array[middle]) {
            right = middle;
        } else if (value > array[middle]) {
            left = middle;
        } else {
            return middle;
        }
        middle = left + ((right - left) >> 1);
    }

    return -1;
}

int _tmain(int argc, _TCHAR* argv[])
{
    int array[] = {2, 4, 7, 11, 23, 57, 89, 46, 90};
    printf("%d\n", binary_search(array, 9, 2));
    printf("%d\n", binary_search(array, 9, 90));
    printf("%d\n", binary_search(array, 9, 57));
    printf("%d\n", binary_search(array, 9, 23));
    printf("%d\n", binary_search(array, 9, 26));
    getchar();
    return 0;
}
```

58楼 [v\\_JULY\\_v](#) 2012-02-10 11:36发表



updated: 各位，可以到此处0积分下载本blog最新博文集锦第6期CHM文件：  
[http://download.csdn.net/detail/v\\_july\\_v/4020172](http://download.csdn.net/detail/v_july_v/4020172)。




Re: [yongjiyu](#) 2012-04-20 08:57发表

中间有好几个连续相同的值时，没有考虑！到底是返回最左边的位置还是最右边的位置，代码是需要稍微修改一下的。回复v\_JULY\_v:

Re: [v\\_JULY\\_v](#) 2012-04-20 09:50发表

回复yongjiyu： 嗯？ 要作何修改？


57楼 [xiaobao\\_code](#) 2012-02-10 11:32发表



尝试写了一下，出现的问题都是楼主说的。佩服佩服啊。  
求资料一份，谢谢楼主，很喜欢你的博客~  
[jxiaobao@gmail.com](mailto:jxiaobao@gmail.com)  
//返回值为int类型，-1表示没有， 则表示出现的位置

```
template<typename T>
int binarySearch(const vector<T> &var, T key) {
int size = var.size();//数组的长度
int mid = (size - 1) / 2;//中间位置
int left = 0, right = size - 1;
while (left <= right) {
if (key < var[mid]) {
right = mid - 1;
} else if (key > var[mid]) {
left = mid + 1;
} else {
return mid;
}
mid = (left + right) / 2;
}
return -1;
}
```

56楼 [jiangbin00cn](#) 2012-02-08 09:17发表



收不了CSDN的评论排版了  
<http://blog.csdn.net/jiangbin00cn/article/details/7241154>

优化：减少主循环的计算量，代价是增大了某一条件下的计算量，总体而言是合算的。  
int middle=left + ((right-left)>>1); //防止溢出，移位也更高效。同时，每次循环都需要更新

另外：在循环内申请变量的习惯不太好，若是多层循环可能导致变量的多次申请和释放。虽然编译器可能作出优化，但这个习惯不太好。

55楼 [jiangbin00cn](#) 2012-02-08 09:01发表



/\*  
对折法

参数：  
DWORD IN\_pArray[], 待查找数组  
DWORD IN\_ArrayNum, 数组大小  
DWORD Value 待匹配数据

返回：  
0xffffffff 查找识别  
0-IN\_ArrayNum-1 匹配成功，返回位置  
\*/

```
DWORD BinarySearch(DWORD IN_pArray[],DWORD IN_ArrayNum,DWORD Value )
{
//首先判断是否在数组中
if( (Value<IN_pArray[0]) || (Value>IN_pArray[IN_ArrayNum-1]) ) return 0xffffffff;

//数组头指针
DWORD *pArray=IN_pArray;
//元素个数
DWORD Num=IN_ArrayNum;
DWORD Temp;
```

//元素个数为1时，无需对折  
while( Num>1 )  
{

```
//数量直接右移
Temp=Num>>1;

//是否在数组右半部
if( pArray[Temp]<Value )
{
pArray+=Temp+1;
Num-=Temp+1;
}
//是否在数组左半部
else if( pArray[Temp]>Value )
{
Num=Temp;
}
//匹配成功
else
{
//返回位置
return pArray+Temp-IN_pArray;
}
}

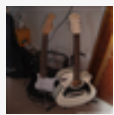
//当元素个数为1是判断是否匹配，返回位置
if( pArray[0]==Value ) return pArray-IN_pArray;
return 0xffffffff;
}
```

54楼 qq511246732 2011-12-30 22:17发表



楼主，小弟菜鸟一个。这句移位运算符等还不能理解通透。  
int middle=left + ((right-left)>>1);  
请问这句代码没有用移位运算符，该怎么实现的？望帮助，不胜感激！

Re: qq511246732 2011-12-30 22:25发表



回复qq511246732：哦哦，就是除二的意思，哈哈

53楼 wangzhiqing3 2011-12-28 10:58发表



```
public class Binary_2 {
    public int binary_search(int arr[],int n,int value){
        int left=0;
        int right=n-1;
        while(left<=right){
            int midle=left+((right-left)>>1);
            if(arr[midle]<value)
                left=midle+1;
            else if(arr[midle]>value)
                midle=right-1;
            else
                return midle;
        }
        return 0;
    }
    public static void main(String args[]){
        int arr[]={1,2,3,4};
        int n=4;
        int value=3;
        Binary_2 b=new Binary_2();
        int t=b.binary_search(arr, n, value);
        System.out.println(t);
    }
}
```

Re: v\_JULY\_v 2011-12-28 11:01发表



回复wangzhiqing3：差不多了，不过，给你指出可能是一笔误：right=middle-1;

52楼 v\_JULY\_v 2011-12-28 10:39发表



引用“v\_JULY\_v”的评论：  
注释里已经说得很明白了，可还是会有不少朋友犯此类的错误：  
[code=cpp]  
//首先要把握下面几...

还有一个最最常犯的错误是@土豆：

middle= (left+right)>>1; 这样的话left与right的值比较大的时候， 其和可能溢出。

51楼 [ghlzs](#) 2011-12-28 10:06发表



int middle = left + ((right - left) >> 1);

为什么这样移位就能防止溢出而且更高效呢， 楼主给解释一下把， 多谢了

Re: [v\\_JULY\\_v](#) 2011-12-28 10:37发表



回复ghlzs： 土豆云： middle= (left+right)>>1; 这样的话left与right的值比较大的时候， 其和可能溢出， 减法可以使得结果为一个right还小的值， 至于移位运算和直接的除法运算， 移位要更快一点。人生得数知己， 死而无憾呐。

50楼 [hnwzs](#) 2011-12-27 14:53发表



凑个热闹请指教

```
int min = 0;
int max = a.Length - 1;
int middle;
while (min <= max)
{
    middle = (max + min) / 2;
    if (num > a[middle])
    {
        min = middle + 1;
    }
    else if (num < a[middle])
    {
        max = middle - 1;
    }
    else
    {
        return middle;
    }
}
return -1;
```

49楼 [v\\_JULY\\_v](#) 2011-12-27 11:54发表



非常感谢所有在本帖留下评论或代码的所有读者，感谢各位。

48楼 [v\\_JULY\\_v](#) 2011-12-27 10:33发表



chunyang xiao发邮件给的代码：

```
#include <iostream>
#include <vector>

using namespace std;

template<typename T>
int _binary_search(const vector<T>& vec, const T& search, const int begin, const int end){

    //Return the index of the search, -1 if the searched value doesn't exist
    //The vec is sorted from small to big

    while(begin <= end){
        int middle = (begin+end) >> 1;
        if (search < vec[middle])
            return _binary_search(vec, search, begin, middle-1);
        if (search > vec[middle])
            return _binary_search(vec, search, middle+1, end);
        else
            return middle;
    }

    return -1;
}

template<typename T>
int binary_search(const vector<T>& vec, const T& search){
    return _binary_search(vec, search, 0, vec.size()-1);
}

int main(){


    vector<int> v_test;
    v_test.push_back(0);
    v_test.push_back(1);
    cout << binary_search(v_test, 1) << endl;
    v_test.push_back(5);
    cout << binary_search(v_test, 1) << endl;
    cout << binary_search(v_test, 12) << endl;


}
```




47楼 [qiulei11111](#) 2011-12-27 09:16发表 

 qdqwd


46楼 [qiulei11111](#) 2011-12-27 09:16发表 

 dqwd

45楼 [qiulei11111](#) 2011-12-27 09:16发表 

 weq

44楼 [niyuan9981](#) 2011-12-27 09:00发表 

 测试过没有发现问题


43楼 [v\\_JULY\\_v](#) 2011-12-26 17:27发表 



注释里已经说得很明白了，可还是会有不少朋友犯此类的错误：

```
//首先要把握下面几个要点：
//right=n-1 => while(left <= right) => right=middle-1;
//right=n   => while(left <  right) => right=middle;
//middle的计算不能写在while循环外，否则无法得到更新。
```

42楼 [niyuan9981](#) 2011-12-26 16:22发表 

 int binary\_search(int array[],int len,int val)

```
{


    int i,s,p;
    p=len-1;
    s=0;

    if(array[s]==val)
    {
        return s;
    }
    else if(array[p]==val)
    {
        return p;
    }

    while(s<(p-1))
    {
        i=(p+s)/2;


        if(array[i]>val)
        {
            p=i;
        }
        else if(array[i]<val)
        {
            s=i;
        }
        else
        {
            return i;
        }
    }
    return -1;
}
```

请指教， 9017902@qq.com

Re: [v\\_JULY\\_v](#) 2011-12-26 16:37发表 



回复niyuan9981： 边界条件出错~。

Re: [niyuan9981](#) 2011-12-26 17:01发表 

回复v\_JULY\_v： 没明白， 什么情况下会出错？



Re: [v\\_JULY\\_v](#) 2011-12-26 17:25发表



回复niyuan9981：何不自己测试一下下？

Re: [niyuan9981](#) 2011-12-26 17:46发表



回复v\_JULY\_v：//首先要把握下面几个要点：

//right=n-1 => while(left <= right) => right=middle-1;

//right=n => while(left < right) => right=middle;

你是说这个吗，我减1是因为我把最边上的值在循环外判断过了，循环内只判断中间值

Re: [v\\_JULY\\_v](#) 2011-12-26 18:06发表



回复niyuan9981：测试了么？

41楼 [gaofengkuiyue](#) 2011-12-26 15:30发表



求楼主一份资料，谢谢。453049664@qq.com

//二分查找的的时候，必须是排好序的数组

```
private static int BinarySearch(int[] arr,int value)
```

```
{
```

```
if(arr.Length>0) //只有数组不为空，才能排序
```

```
{
```

```
int left = 0;
```

```
int right = arr.Length - 1;
```

```
int middle = (left + right + 1)/2;
```

```
while (left <= right)
```

```
{
```

```
if (value < arr[middle])
```

```
{
```

```
right = middle - 1;
```

```
}
```

```
else if (value > middle)
```

```
{
```

```
left = middle + 1;
```

```
}
```

```
else
```

```
{
```

```
return middle;
```

```
}
```

```
middle = (left + right + 1) / 2;
```

```
}
```

```
}
```

```
return -1;
```

```
}
```

Re: [v\\_JULY\\_v](#) 2011-12-26 16:01发表



回复gaofengkuiyue：middle不必两次赋值。

40楼 [xx](#) 2011-12-26 14:55发表



我要歪楼了，我承认我现在给我几个小时都写不出正确的二分算法。

因为我知道5/2,余数是1，但是我并不知道为什么1+1=2，我也不需要写代码去实现1+1=? 这个问题。

我认为，在某些领域，算法可以被放到神的位置供奉都不为过，但在大量的应用软件面上，算法只不过是封装在这种那种包或类或其他载体中的对程序员透明的代码。

在晋升高层次的程序员时，或者在进行某些创造性较强的开发工作中，算法可能也会发挥出重要的作用或者成为难以逾越的绊脚石，但是如果以我现在所处的境地，让我现在去搞清楚弄明白那些算法，甚至自己去完整无误的实现一次，我拒绝，虽然我偶尔会在某时某刻特别渴望得知某种算法的实现，偶尔迸发出特别强烈需要改进某种算法的想法。

Re: [v\\_JULY\\_v](#) 2011-12-26 16:03发表



回复njnu06040715：呵呵。只说一点，这里：“我承认我现在给我几个小时都写不出正确的二分算法”，你应该充满自信一点。

39楼 [JackCousins](#) 2011-12-26 14:17发表



"你能正确无误的写出二分查找代码么？" "so easy!妈妈再也不用担心我的学习"



看来我的最麻烦了，请楼主发我一份材料


邮箱地址：xigongda200608@163.com

```
bool QuickSort::Binary_Search(int arr[],int iStart,int iEnd,int iValue,int& index){
/*****/

/*
iStart:查找起始位置
iEnd:查找结束位置
iValue:要查找的数值
index:如果找到，返回该数值所在位置的索引
如果找到，函数返回true，否则返回false
*/

/*****/

if(iStart>iEnd)
{
return false;
}
int midIndex=(iStart+iEnd)/2;
int iMid=arr[midIndex];
if(iMid==iValue)
{
index=midIndex;
return true;
}
if(iMid>iValue)
{
return Binary_Search(arr,iStart,midIndex-1,iValue,index);
}
if(iMid<iValue)
{
return Binary_Search(arr,midIndex+1,iEnd,iValue,index);
}
return false;
}
```

Re: [v\\_JULY\\_v](#) 2011-12-26 11:48发表 



回复KWSY2008： 错的地方太多niao

37楼 [searchwenjie](#) 2011-12-26 00:11发表 



```
//function discription: binary search
//paramiter discription:
// arr:the array to search and be sort in descent
// len:the length of arr
// value:the value to be search
int binary_search(int arr[],int len,int value)
{
int low,high,mid = -1;
int i;

//init search area
low = 0;
high = len - 1;

while(low <= high){
mid = (low + high) / 2;

if( value < arr[mid] ){
high = mid - 1;
}
else if( value > arr[mid] ){
low = mid + 1;
}
else{
return mid;
}
}
return -1;
```



}

36楼 [quanben](#) 2011-12-25 11:51发表



最近经常用ADA写程序，所有用ADA语言写了（类似pascal），凑个热闹，程序期望给出找到位置或插入点位置。  
写了一段时间，还是不太确定是否正确。测试了两个序列，看上去没错。

```
procedure binarysearch(l : list_t; first, last : index_t; e : element_t;
                        index : out index_t; found : out boolean) is
    lo : index_t := first; -- 下界索引
    hi : index_t := last + 1; -- 上界索引 (+1)
    mi : index_t; -- 中间索引
    vmi : element_t; -- 中间值
begin
    while lo < hi loop
        mi := (lo + hi) / 2;
        vmi := get(l, mi);

        if e < vmi then
            hi := mi;
        elsif vmi < e then
            lo := mi + 1;
        else -- vmi = e, found
            index := mi;
            found := true;
            return;
        end if;
    end loop;
    index := lo;
    found := false;
end;
```

35楼 [吕海鹏](#) 2011-12-25 10:18发表



标题扣得帽子有些大 90%却也不太过分，  
这有些责任是老师的事，  
我现在在程序流程训练的基础课程中每次都加上这个练习的，

34楼 [newpb80](#) 2011-12-25 00:09发表



我觉得算法对提高软件开发人员的基础知识还是不错的。  
但是做应用软件开发人员 很多人 算法写不了，但能开发大型应用软件系统的 大有人在。  
关键是提高设计能力和业务知识。

33楼 [owen\\_ccc](#) 2011-12-24 22:59发表



```
#include <stdio.h>

int binary_search(int array[], int size, int value)
{
    int left = 0;
    int right = size - 1;

    while (left <= right) {
        //运算符优先级问题，必须加括号、
        int middle = left + ((right - left) >> 1);

        if (array[middle] < value) {
            left = middle + 1;
        } else if (array[middle] > value) {
            right = middle - 1;
        } else {
            return middle;
        }
    }

    printf("Error! value: %d not exist!\n", value);
    return -1;
}

int main()
{
    int array[10] = {7, 6, 9, 2, 3, 4, 8, 5, 0, 10};
    printf("Position: %d\n", binary_search(array, 10, 5));
    return 0;
}

// Email : owen_ccc#163.com
```

32楼 xni220 2011-12-24 17:21发表



看了楼主的帖子，蛮感兴趣的，可以发一份资料给我吗？谢谢。

Re: xni220 2011-12-24 17:22发表



回复xni220： 忘记写上邮箱了xni220@126.com

31楼 lzj509649444 2011-12-24 13:56发表



楼主看过这篇“二分查找学习札记”吗，里面总结的比较好！<http://blog.csdn.net/lzj509649444/article/details/7064203>

至于测试的话，10楼的有些观点还是值得思考学习的，如果非要那么全面的测试，应该是测试人员的工作。bug的存在，往往是自己代码有漏洞，楼主更应当思考展示几个典型的测试用例，普通的测试用例也可以贴出来！

30楼 completly 2011-12-24 13:18发表



麻烦lz把资料发我一份吧。419049501@qq.com。谢谢拉。

```
#include<stdio.h>

int binary_search(int data[],int size,int key)
{
    int min=0,max=size-1;
    int middle;
    while(min<=max)
    {
        middle=min+((max-min)>>1);
        if(key>data[middle])
        {
            min=middle+1;
        }
        else if(key<data[middle])
        {
            max=middle-1;
        }
        else
            return middle;
    }

    return -1;
}

int main()
{
    int a[10]={1,2,3,4,5,6,7,8,9,10};
    int key;
    printf("enter an integer(1-10):\n");
    scanf("%d",&key);
    printf("the position of key is:%d\n",binary_search(a,10,key));
    return 0;
}
```

29楼 cjoooooo 2011-12-24 12:53发表



我是来看排序的。没有。。。。。。。。。


28楼 Jocodeoe 2011-12-24 10:34发表




```
int Search(int array[],int n,int value)
{
    int min=0;
    int max=n;
    int temp;
    while(1)
    {
        temp=(min+max)/2;
        if(array[temp]>value)
            max=temp;
        else if(array[temp]<value)
            min=temp;
        else
            return temp;
    }
}
```

之前写得不太好，修改一下。


27楼 [longlong530](#) 2011-12-24 08:34发表

 这个有意义，马上写个试试

26楼 [quanben](#) 2011-12-24 07:59发表


 二分查找是最常用算法之一，原理简单，难在边界条件，很经典

25楼 [shangzhihaohao](#) 2011-12-23 20:14发表


 引用“Jocodeoe”的评论：  
[code=cpp]  
int Search(int array[],int n,int value)...

如果查找失败你就悲剧了。

Re: [Jocodeoe](#) 2011-12-23 22:00发表

 回复shangzhihaohao：粗略地测试了一下，没有发现问题~


24楼 [Jocodeoe](#) 2011-12-23 18:16发表



```
int Search(int array[],int n,int value)
{
    int min=0;
    int max=n;
    int temp;
    while(temp=(min+max)/2)
    {
        if(array[temp]==value) return temp;
        else if(array[temp]>value)
        {
            max=temp;
            continue;
        }
        else
            min=temp;
    }
}
```

请LZ审查~邮箱：109734103@qq.com

23楼 [卡卡\\_苏米](#) 2011-12-23 16:13发表


 我上次有叫我当场写出快速排序算法的，结果就。。。

22楼 [LaoLiulaoliu](#) 2011-12-23 15:37发表

 #python 小测一下，发现真不容易一次写对。烦请楼主帮忙看看

```
def binary_search(n, arr, arr_len=None):
    """n can be numbers,characters,strings...
    arr is n type list """
    if arr_len is None:
        arr_len = len(arr)
    low, high = 0, arr_len - 1
    while True:
        if low >= high:
            return None
        mid = int((low + high) / 2)
        if arr[mid] < n:
            low = mid
        elif arr[mid] > n:
            high = mid
        else:
            return mid
```

Re: [v\\_JULY\\_v](#) 2011-12-23 15:42发表

 回复LaoLiulaoliu：边界值调整：

```
//首先要把握下面几个要点：
//right=n-1 => while(left <= right) => right=middle-1;
//right=n   => while(left <  right) => right=middle;
//middle的计算不能写在while循环外，否则无法得到更新。
```

Re: [LaoLiulaoliu](#) 2011-12-23 22:41发表

回复v\_JULY\_v：还真是有问题，谢谢楼主



21楼 [hellorld](#) 2011-12-23 14:25发表



仅就本程序而言，因为left = 0,所以 (left + right) / 2 也不会溢出吧。

Re: [miraclecoder](#) 2014-04-10 16:02发表



回复hellorld：但是你想一下，left是会变化的，最简单的猜想：第一次left赋与middle值，则left就马上增大。不妨假设right很大，但是int仍不举出，但是下一次再求middle则有可能溢出了

20楼 [skylkj](#) 2011-12-23 11:01发表



c的，5年前写的一个。原本程序是二分法找索引，然后返回内容。所以找到后返回的下标都是+1的

```
while(1)
{
if (datain == char_pointer[low*2])
return char_pointer[low*2+1];
else if (datain == char_pointer[hig*2])
return char_pointer[hig*2+1];
else if (hig-low <= 1)
{
//printf("low:%d hig:%d \r\n",low,hig);
return 0;
}
```

```
temp = (low + hig)/2;
if (datain==char_pointer[temp*2])
return char_pointer[temp*2+1];
```

```
if (datain<char_pointer[temp*2])
hig = temp;
else
low = temp;
}
```

邮箱 ylagra@163.com 谢谢

19楼 [hotpos](#) 2011-12-23 09:50发表



```
int bin_search(int arr[], int n, int val)
{
    int l = 0, r = n - 1;

    if (val == arr[r])
        return r;
    else if (val > arr[r])
        return -1;

    if (val == arr[l])
        return l;
    else if (val < arr[l])
        return -1;

    while (l < r)
    {
        int p = l + (r-l)/2;

        if (p == l || p == r)
            break;

        if (arr[p] == val)
            return p;
        else if (arr[p] > val)
            r = p;
        else
            l = p;
    }

    return -1;
}
```

发现一个BUG。。

xujiafea123@gmail.com 感谢博主分享

Re: [v\\_JULY\\_v](#) 2011-12-23 09:58发表



回复hotpos：既然发现了（好几处）， 还不改过？

18楼 [西部流云](#) 2011-12-23 09:15发表



```
Java版本：
public static int bisearch(int [] arr , int n){
int low = 0;
int height = arr.length -1;
if(arr[low] == n) return low;
if(arr[height] == n) return height;

while(low <= height){
int mid = (low+height)/2;
if(arr[mid] == n) return mid;
else if(arr[mid] > n){
height = mid-1;
}
else{
low = mid+1;
}
}
return -1;
}
```

17楼 [complety](#) 2011-12-22 23:24发表



```
#include<stdio.h>
int binary_search(int data[],int size,int key)
{
int min=0,max=size-1;
int middle=min+((max-min)>>1);
if(data[max]==key)
return max;
while(min<=max)
{
if(key>data[middle])
{
min=middle;
middle=min+((max-min)>>1);
}
else if(key<data[middle])
{
max=middle;
middle=min+((max-min)>>1);
}
else
return middle;
}

return -1;
}

int main()
{
int a[10]={1,2,3,4,5,6,7,8,9,10};
int key;
printf("enter an integer(1-10):\n");
scanf("%d",&key);
printf("the position of key is:%d\n",binary_search(a,10,key));
return 0;
}

//刚刚写的时候忘了while里面查不出来最后的那个数，于是在开头就加了个if，419049501@qq.com这个是我邮箱，谢谢lz把资料共享，嘿嘿
```

16楼 [aikaibo](#) 2011-12-22 19:17发表



```
public class ArrayTwainLookup
{
public static int lookup(int[] array, int value)
{
int low = 0;
```

```
int height = array.length - 1;
while(low <= height)
{
int i = (low / 2) + (height / 2);
if(array[i] == value)
{
return i;
}
else if(array[i] < value)
{
low = i + 1;
}
else
{
height = i - 1;
}
}
return -1;
}


public static void main(String[] args)
{
int[] array = new int[]{1,3,4,8,9,12,34};

int s = lookup(array,10);

System.out.println(s);
}


}
```

这是我写的呵呵 我的邮箱 414398175@qq.com

15楼 [快乐的海盗](#) 2011-12-22 19:07发表 




LZ可否把经典算法发我一份，让我学习一下，非常感谢。cjxvswf@sina.com

14楼 [wangxzi](#) 2011-12-22 18:00发表 




写了一下，总能测出bug，跳了半个小时才搞定，好惭愧！ 楼主态度技能皆值得尊重。 希望楼主能发一份学习资料 general\_use@163.com 谢谢！

Re: [v\\_JULY\\_v](#) 2011-12-22 18:05发表 




回复wangxzi： 可以，不过你先要把你的代码贴出来给我看一下。:)

13楼 [xug12345](#) 2011-12-22 17:41发表 



有点儿小越界

```
int search(int a[],int n,int v)
{
if(a[0]==v)return 0;
if(n==1)
return a[0]==v?-1;
int i=n>>1;
if(a[i]==v)
return i;
else if(a[i]>v)
return search(a,i,v);
else
{
int x=search(a+i+1,i,v);
return x<0?x:x+i+1;
}
}
```

Re: [xug12345](#) 2011-12-23 10:29发表 



xug12345@hotmail.com

12楼 [meichen8050753](#) 2011-12-22 16:47发表



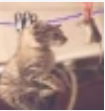
没有什么准确的。。

11楼 [cheniwantyou](#) 2011-12-22 11:46发表



你写出了二分查找，然后你呢？代表什么？很牛逼吗？还是满足你意淫的需要？

Re: [吹泡泡的小猫](#) 2011-12-22 22:34发表



回复cheniwantyou：真想研究一下你这种人的心理

Re: [yikecai](#) 2011-12-22 17:13发表



回复cheniwantyou：你个煞笔你也写一个出来，证明楼主能做到的是你不能，楼主是牛逼，你是傻逼

Re: [v\\_JULY\\_v](#) 2011-12-22 12:21发表



回复cheniwantyou：不代表任何什么，朋友。

10楼 [alex\\_yu\\_e](#) 2011-12-22 10:27发表



另外看了楼主这么多篇博文，您的测试方法也让我很想吐槽，无论一个多么复杂的算法，测试数据就一个大小不超过10的数组，至于这个算法的效率，到底有多高，以及这个程序运行时的内存分配、运行时间情况，一概没有测试，您的层次只是到把它实现出来而已，既然您都花了这么大工夫把它写出来了，多给点测试数据，看看程序运行时的内存分配、时间，调试测试一下然后告诉我们结果又有何妨

9楼 [alex\\_yu\\_e](#) 2011-12-22 10:00发表



另外，楼主的不调程序直接贴出来也让我很汗颜，不调试测试的程序能叫程序吗，这样直接贴出来除了是大家开开玩笑而已有什么意义呢

8楼 [alex\\_yu\\_e](#) 2011-12-22 09:55发表



楼主这篇博文让我想起了孔乙己的回字有多少种写法，指导一下在学数据结构的学生还行，但单单拿会写一个正确的二分查找来说自己比多少多少专业程序员强就有点那个了，毕竟程序员的世界不全是算法和数据结构，即使一时半会写不出来正确的，网上随便找一个或者参考您的总可以吧

Re: [richardzrc](#) 2014-05-06 11:58发表



回复alex\_yu\_e：从我的理解看，如果你什么都是网上抄一段，当然可以，但是当自己设计一个新的算法时候，就不一样了。国内90%以上程序员都是低级码农，当你已经到了这个地步，想要突破的时候，就需要算法能力来提升为google一样的高级程序员了：)

7楼 [yangwp521](#) 2011-12-22 09:33发表



```
private int search(int[] a, int f, int r, int value) { //f表示数组头部，r表示尾部
    while(f <= r) {
        int temp = (r + f) >> 1;
        if(value > a[temp]) {
            f = temp + 1;
            search(a, f, r, value);
        }
        if(value < a[temp]) {
            r = r >> 1 - 1;
            search(a, f, r, value);
        }
        if(value == a[temp])
            return temp + 1;
    }
    return 0;
}
```

6楼 [a8326369](#) 2011-12-22 08:52发表



int middle=left + ((right-left)>>1); //防止溢出，移位也更高效。同时，每次循环都需要更新。

这里，不明白你说的“高效”是什么意思。

你这样计算，需要算1次移位和2次加法。虽然可以有效的防止溢出，但是称不上高效。

《代码之美》介绍过一个高效的算法

int middle = left + right >>> 1; //无符号移位

这样只需要1次加法一次移位就可以了

Re: [a8326369](#) 2011-12-22 08:59发表



回复a8326369：哦，不好意思我浅薄了，原来C++没有无符号移位。。。。基础不牢，见谅

5楼 [michaelscofielddong](#) 2011-12-21 21:00发表





```
#include <iostream>
```

```
using namespace std;
```

```
int binSearch(int a[],int data,int begin, int end)
```

```
{  
int mid = 0;
```

```
while(begin <= end)  
{  
mid = (begin + end)/2;  
if(a[mid] > data)  
{  
end = mid - 1;  
}  
else if(a[mid] < data)  
{  
begin = mid + 1;  
}  
else  
{  
return mid;  
}  
}  
return -1;  
}
```

```
int main(void)  
{  
return 0;  
}
```

Re: [v\\_JULY\\_v](#) 2011-12-21 21:05发表



回复michaelscofielddong: 边界值未初始化, 主函数中未调用binSearch

4楼 [longwen0124](#) 2011-12-21 20:53发表



//返回要查找的数在数组中的位置, 没有找到则放回-1  
//数组中的元素是升序

```
int binary_search(int array[], int n, int value)  
{  
int left = 0,right = n-1;  
int mid;
```

```
while(left <= right)  
{  
mid = (left + right)/2; //每次取中间位置
```

```
if(array[mid] < value) //value在mid位置后面  
{  
left = mid + 1;  
}  
else if(array[mid] > value) //value在mid位置前面  
{  
right = mid - 1;  
}  
else //不大于, 不小于就只能等于了咯, 等于就找到了  
{  
return mid;  
}  
}  
return -1; //没有找到返回-1  
}
```

3楼 [wangeryuanbj](#) 2011-12-21 20:51发表



```
int BSearchValue(int arr[], int n, int nValue)  
{  
int tmpValue = 0;
```

```

int nFirst = 0;
int nLast = n - 1;
int nMid = 0;

while (nFirst <= nLast)
{
    nMid = (nFirst + nLast)/2;
    (arr[nMid] < nValue) ? (nFirst = nMid + 1) : (arr[nMid] > nValue ? nLast = nMid - 1 : tmpVa
    if (tmpValue != 0) return tmpValue;
}
return -1;
}

int _tmain(int argc, _TCHAR* argv[])
{
    int arr[10] = {1,2,3,4,5,6,7,8,9,10};
    int nMid = BSearchValue(arr, 10, 6);
    cout<<nMid<<endl;
    for (;;) {}
    return 0;
}

```

2楼 [zhucegepp](#) 2011-12-21 20:44发表



```

import java.util.ArrayList;

public class BinarySearch {

    public static int binarySearch(ArrayList<Integer> numList, int value) {
        int left = 0;
        int right = numList.size() - 1;
        while (left <= right) {
            int middle = (left + right) >> 1;
            if (numList.get(middle) == value) {
                return middle;
            } else if (numList.get(middle) > value) {
                right = middle - 1;
            } else {
                left = middle + 1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {

        ArrayList<Integer> numList = new ArrayList<Integer>();
        for (int i = 0; i < 10; i++) {
            numList.add(i);
        }
        System.out.println(binarySearch(numList, 5));
    }

}

```

1楼 [hljhnu](#) 2011-12-21 20:40发表



```

#include <stdio.h>
#include <stdlib.h>

int bina_search(int data[],int size,int key)
{
    int left=1,right=size;
    while(left<=right)
    {
        int middle=(left+right)/2;
        int mid_value=data[middle];
        if(key==mid_value)return middle;
        else if(key<mid_value) right=middle-1;
        else left=middle+1;
    }
    return -1;
}

int main()
{
    int data[5]={0,1,4,5,7};
    int index=bina_search(data,4,7);
    printf("%d",index);
    system("pause");
    return 0;
}

```

Re: [v\\_JULY\\_v](#) 2011-12-21 20:50发表



回复hljhnu：代码没有考虑是否溢出的问题







您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场






核心技术类目

全部主题    Hadoop    AWS    移动游戏    Java    Android    iOS    Swift    智能硬件    Docker    OpenStack  
VPN    Spark    ERP    IE10    Eclipse    CRM    JavaScript    数据库    Ubuntu    NFC    WAP    jQuery  
BI    HTML5    Spring    Apache    .NET    API    HTML    SDK    IIS    Fedora    XML    LBS    Unity  
Splashtop    UML    components    Windows Mobile    Rails    QEMU    KDE    Cassandra    CloudStack    FTC  
coremail    OPhone    CouchBase    云计算    iOS6    Rackspace    Web App    SpringSide    Maemo  
Compuware    大数据    aptech    Perl    Tornado    Ruby    Hibernate    ThinkPHP    HBase    Pure    Solr  
Angular    Cloud Foundry    Redis    Scala    Django    Bootstrap

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

 网站客服    杂志客服    微博客服    webmaster@csdn.net    400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved   

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

 网站客服    杂志客服    微博客服    webmaster@csdn.net    400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持  
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved   